

Chapter 14 - Count Models

Supplemental Code

Brian Fogarty

This document provides the code for creating curvilinear predicted count plots and for running a zero-inflated negative binomial regression model (from the *Comparing Zero-Inflated, Hurdle, and Standard Count Models* subsection of the *Zero-Inflated & Hurdle Count Models* section).

Curvilinear Predicted Count Plots

```
library(tidyverse)
citations <- read_csv("citations_twitter.csv")
glimpse(citations)
```

```
Rows: 308
Columns: 7
$ articlecites    <dbl> 8, 0, 13, 1, 2, 14, 1, 10, 6, 3, 13, 12, 3, 3, 14, 7, ~
$ womanleadauthor <dbl> 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, ~
$ womanauthor     <dbl> 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, ~
$ tweet_dum       <dbl> 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, ~
$ fullprof        <dbl> 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, ~
$ retweets        <dbl> 2, 0, 2, NA, NA, 0, NA, 35, 0, NA, 3, 5, NA, NA, 0, 1, ~
$ totfav          <dbl> 6, 0, 7, NA, NA, 1, NA, 64, 1, NA, 8, 5, NA, NA, 1, 2, ~
```

The code for curvilinear predicted count plots is very similar to creating predicted probability plots. We need to have an ordinal, interval, or ratio-level variable on the x -axis. For this plotting example, we'll run a simple negative binomial model with `articlecites` as the outcome variable and `retweets` and `womanleadauthor` as predictors.

```
library(MASS)
summary(model.nbrm <- glm.nb(articlecites ~ retweets + womanleadauthor, data = citations))
```

Call:

```
glm.nb(formula = articlecites ~ retweets + womanleadauthor, data = citations,
  init.theta = 1.388327942, link = log)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.0871	-0.8966	-0.2964	0.4421	1.7664

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.599521	0.146543	10.915	<2e-16 ***
retweets	0.007487	0.005905	1.268	0.205

```
womanleadauthor 0.063768 0.225014 0.283 0.777
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.3883) family taken to be 1)

Null deviance: 87.111  on 74  degrees of freedom
Residual deviance: 85.399  on 72  degrees of freedom
(233 observations deleted due to missingness)
AIC: 420.12

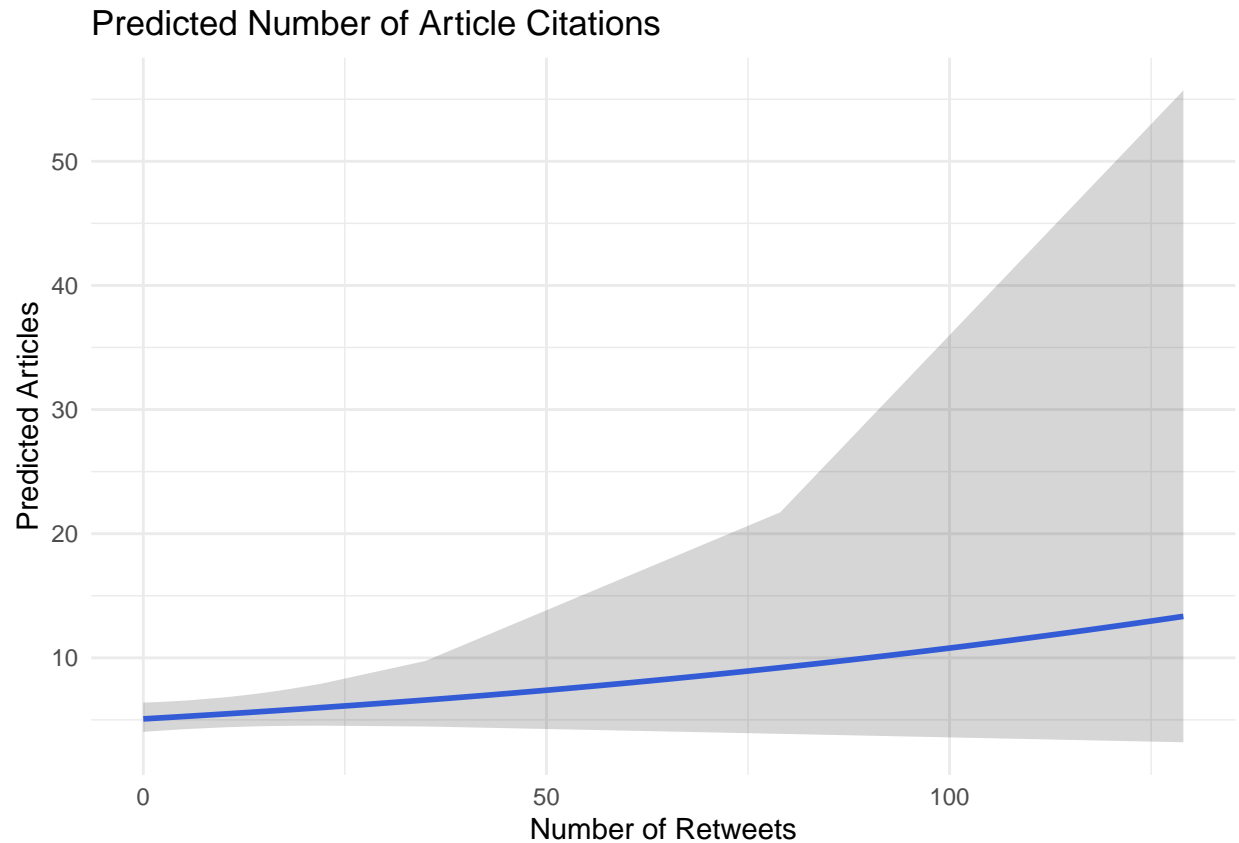
Number of Fisher Scoring iterations: 1

      Theta: 1.388
    Std. Err.: 0.303

2 x log-likelihood: -412.123
```

Neither predictor is statistical significant, which will make the predicted counts plot pretty ugly.

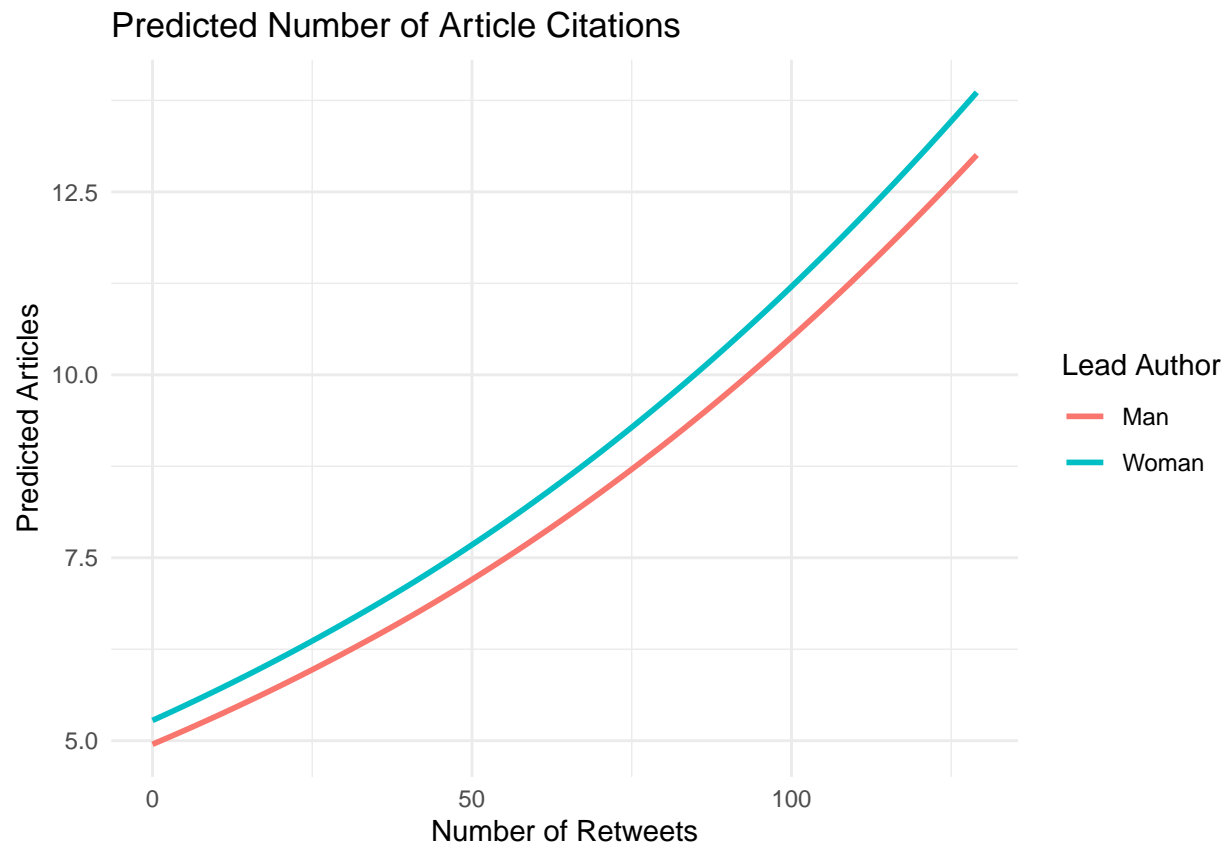
```
library(ggeffects)
ggpredict(model.nbrm, terms = "retweets") %>%
ggplot(mapping = aes(x = x, y = predicted)) +
  geom_smooth(se = FALSE) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .2) +
  labs(title = "Predicted Number of Article Citations",
       x = "Number of Retweets", y = "Predicted Articles") +
  theme_minimal()
```



Yeah, very ugly - especially with the confidence/prediction intervals. We might want drop the confidence/prediction intervals by removing the `geom_ribbon()` function.

We can split the predicted counts by `womanleadauthor`; we'll also cut out the confidence/prediction intervals.

```
ggpredict(model.nbrm, terms = c("retweets", "womanleadauthor")) %>%
  mutate(group = as_factor(group),
         group = recode(group,
                        `0` = "Man",
                        `1` = "Woman")) %>%
  ggplot(mapping = aes(x = x, y = predicted, colour = group)) +
  geom_smooth(se = FALSE) +
  labs(title = "Predicted Number of Article Citations",
       x = "Number of Retweets", y = "Predicted Articles") +
  guides(colour = guide_legend(title = "Lead Author")) +
  theme_minimal()
```



Zero-Inflated Negative Binomial Model

The below continues the discussion found in the chapter on zero-inflated count models in the *Comparing Zero-Inflated, Hurdle, and Standard Count Models* subsection. Let's first re-run the standard negative binomial model and re-check for excess zeros.

```
summary(model.nbrm <- glm.nb(articlecites ~ tweet_dum + womanleadauthor +
                             fullprof, data=citations))
```

Call:

```
glm.nb(formula = articlecites ~ tweet_dum + womanleadauthor +
       fullprof, data = citations, init.theta = 1.097281755, link = log)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.0402	-1.5328	-0.4058	0.3874	3.2615

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.74365	0.10388	7.159	8.13e-13 ***
tweet_dum	0.84316	0.14326	5.885	3.97e-09 ***
womanleadauthor	0.04831	0.13419	0.360	0.719
fullprof	0.19177	0.13277	1.444	0.149

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.0973) family taken to be 1)

Null deviance: 381.90 on 305 degrees of freedom

Residual deviance: 341.81 on 302 degrees of freedom

(2 observations deleted due to missingness)

AIC: 1359.5

Number of Fisher Scoring iterations: 1

Theta:	1.097
Std. Err.:	0.139

2 x log-likelihood: -1349.518

```
library(performance)
check_zeroinflation(model.nbrm)
```

Check for zero-inflation

Observed zeros:	81
Predicted zeros:	78
Ratio:	0.96

Model seems ok, ratio of observed and predicted zeros is within the tolerance range.

Now, we'll run a ZINB model using the `zeroinfl()` function, from the `pscl` package, by including the specification `dist = "negbin"`.

```
library(pscl)

summary(model.zinb <- zeroinfl(articlecites ~ tweet_dum + womanleadauthor +
                                fullprof | tweet_dum + womanleadauthor +
                                fullprof, data=citations, dist = "negbin"))
```

Call:

```
zeroinfl(formula = articlecites ~ tweet_dum + womanleadauthor + fullprof |
          tweet_dum + womanleadauthor + fullprof, data = citations, dist = "negbin")
```

Pearson residuals:

	Min	1Q	Median	3Q	Max
	-1.0188	-0.7375	-0.3705	0.4445	6.9698

Count model coefficients (negbin with log link):

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.8052	0.1202	6.701	2.07e-11 ***
tweet_dum	0.7711	0.1440	5.354	8.58e-08 ***
womanleadauthor	0.1526	0.1403	1.088	0.277
fullprof	0.1134	0.1376	0.824	0.410
Log(theta)	0.2169	0.1517	1.430	0.153

Zero-inflation model coefficients (binomial with logit link):

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-3.637	3.343	-1.088	0.277
tweet_dum	-10.018	101.260	-0.099	0.921
womanleadauthor	2.422	3.287	0.737	0.461
fullprof	-10.585	109.452	-0.097	0.923

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 1.2422
Number of iterations in BFGS optimization: 45
Log-likelihood: -672.2 on 9 Df

Do we see anything strange about the results? Yep, the standard errors (and coefficients) for `tweet_dum` and `fullprof` in the inflation model are ginormous. Since the inflation model directly informs the estimation of the count model in zero-inflated models, these huge standard errors suggest the presence of multicollinearity. The multicollinearity observed here is likely due to how well the NBRM performs in predicting the number of zeros in `articlecites`.¹ Therefore, the predictors in this inflation model are trying to duplicate the job that the predictors in the NBRM already have covered; the NBRM doesn't need any help from the inflation model for dealing with zeros.

¹The multicollinearity in this situation is a bit different from what we discussed for linear regression models in Chapter 12.